INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 18, Number 2, Pages 190–202

FINITE VOLUME ELEMENT METHOD FOR PREDICTING ELECTROSTATICS OF A BIOMOLECULE IMMERSED IN AN IONIC SOLVENT

HAO WU, JINYONG YING*, AND QINGSONG ZOU

Abstract. Poisson-Boltzmann equation (PBE) is a classic implicit continuum model to predict the electrostatic potentials of a solvated biomolecule. In this paper, we present a finite volume element method specific to the elliptic interface problem with a non-homogeneous flux condition for solving PBE and provide a follow-up analysis. The new PBE solver is fulfilled through both `Fortran` and `Python`, afterwards the local Poisson test model coupled with an analytical solution is adopted to well validate the program. Lastly, an application of the new solver to the prediction of solvation free energies of the proteins is made.

Key words. Poisson-Boltzmann equation, electrostatic free energy, finite volume element method, solvation free energy.

1. Introduction

The electrostatics referring to a protein immersed in an ionic solvent are important to recognize its biological structure and the various relevant functions [22, 26]. At present, one commonly-used mathematical model for predicting electrostatics is the Poisson-Boltzmann model, which has been employed in various applications such as protein docking, ion channel modeling, and rational drug design [21]. Up to now, the mathematical theory of Poisson-Boltzmann equation (PBE) and its variants have been well analyzed [14, 29] by considering an electrostatic free energy minimization problem subject to the Poisson dielectric model. Meanwhile, these models have been efficiently and accurately solved by finite element method (FEM) [1, 12, 15, 28], finite difference method [27, 38], boundary element method [19], and some mixed methods [2, 34, 35, 36]. Besides, to simulate the electrodiffusion in numerous biological processes, Poisson-Nernst-Planck equation (PNP) as well as its improved models [20, 23] have also been proposed and commonly used as well.

It is well-known that FEM is used to solve the interface problems of both PBE and PNP thanks to its flexibility of handling the complex interface. As alternatives, finite volume methods not only can deal with the complex interface very well, but also preserve the local conservation laws of some physical quantities such as mass and flux. Additionally, in comparison with FEM, its computational cost is relatively less while it aims at the explicit evolved equation with time. To authors' best knowledge, it only has been used so far to solve on Cartesian grids both PBE [10] and size-modified PBE [24] without explicitly considering the interface, and to predict the double layer forces between spherical colloidal particles [17], latter of which considered the Poisson equation without interface. Therefore, finite volume methods have never been used to solve PBE and its variants yet. The finite volume element method (FVEM, see e.g. [3, 11, 16, 33, 37]), one type of finite volume methods, has gained increasing attention recently. Owe to the same mesh

Received by the editors August 29, 2020 and, in revised form, January 6, 2021.
2000 *Mathematics Subject Classification.* 65M08, 92-08.
*Corresponding author.

and basis functions being used in the discretization process as that of FEM, FVEM possesses almost all the advantages of FEM such as flexibility to handle the complex interface and domain with arbitrary geometry. Meanwhile, the preservation of local conservation laws about certain physical quantities makes FVEM especially important for PNP [20]. As the first step of several potential subsequent works, we attempt to solve PBE via FVEM in this work.

By virtue of the decomposition scheme intended for isolating the singularities caused by the Dirac delta distributions, the solution u of PBE is splitted into three parts: solution G in an analytical expression, solution Ψ subject to a linear interface problem with a non-homogeneous flux condition on the interface, and solution $\tilde{\Phi}$ tied to a nonlinear interface problem. Through literature on finite volume methods, an elliptic equation with a non-homogenous flux condition on the interface has never been discussed yet. Our work is the first to propose a new technique to overcome the difficulties induced by the non-homogenous flux condition. For any vertices on the interface in an given unstructured mesh, we artificially separate their control volumes into pairs so that the interface lies on the common boundary of the two sub-control volumes. As a result, the non-homogeneous flux condition can be incorporated into the variational form through integration-by-part performed on those separated sub-control volumes.

Based on the proposed technique, we formulate a new finite volume element PBE solver and fulfill it in both `Python` and `Fortran`. The local Poisson test model owning an analytical solution in a series form involving Legendre polynomials is used to validate the new program. The tests show that applying our technique to deal with the non-homogeneous flux condition gives rise to the second-order convergence in L_2 norm and the first-order convergence in H^1 norm, which is exactly the same as FVEM has achieved [33]. As an application, the new solver is subsequently applied to predict the solvation free energies of some proteins. Meanwhile, the obtained energies are compared to the ones derived from the finite element PBE solver [28] engaging the same unstructured meshes. These numerical tests illustrate that the predicted solvation free energies are quite close to each other although different numerical methods are adopted.

The rest of the paper is organized as follows. A short review of PBE solution decomposition is given in Section 2. Section 3 is devoted to presenting our FVEM formulation specific to the regularized PBE. At the end of this section, a new algorithm for solving PBE is presented. We present an analysis of the FVEM for PBE in Section 4. The validation test on local Poisson test model as well as the application of the new solver in predicting the solvation free energy is conducted in Section 5.

2. Solution decomposition of the Poisson-Boltzmann model

Let Ω be a sufficiently large bounded domain of $\mathbb{R}^3$ (See Figure 1 for an illustration) satisfying

$$\Omega = D_p \cup D_s \cup \Gamma,$$

where D_p denotes a solute region hosting a protein molecule with n_p atoms, D_s denotes a solvent region, and Γ is the interface between D_p and D_s. Under the implicit solvent approach, both D_p and D_s are treated as continuum media with dielectric constants ϵ_p and ϵ_s, respectively. Then for a symmetric 1:1 ionic solvent (e.g., a salt solution with sodium (Na^+) and chloride (Cl^-) ions), the electrostatic potential u (in unit k_BT/e_c) can be predicted by the following boundary value

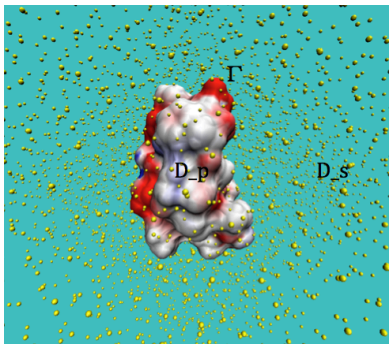

FIGURE 1. An illustration of domain partition for a protein with PDB ID 2LZX immersed in an ionic solvent. Here the yellow dots denote the mobile ions in the solvent.

problem

$$
(1)\qquad \begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) + \kappa^2 \sinh(u) = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s \dfrac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p \dfrac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = g(\mathbf{s}), & \mathbf{s} \in \partial\Omega, \end{cases}
$$

where $\mathbf{r}_j$ and z_j denote the position and charge number of the jth atom, respectively, g is a given boundary function, $\partial\Omega$ is the boundary of Ω, $\delta_{\mathbf{r}_j}$ is the Dirac delta distribution at point $\mathbf{r}_j$, $\mathbf{n}(\mathbf{s})$ is the unit outward normal vector of D_p, and α, κ and M are three physical constants. Under the SI (Le Système International d$'$Unités) units, for a length measured in angstroms (Å), the constants α and κ^2 can be computed by

$$
(2)\qquad \alpha = \frac{10^{10} e_c^2}{\epsilon_0 k_B T}, \quad \kappa^2 = 2 I_s \frac{10^{-17} N_A e_c^2}{\epsilon_0 k_B T},
$$

here e_c, ϵ_0, k_B, T, and N_A are the electron charge, the permittivity of vacuum, the Boltzmann constant, the absolute temperature, and the Avogadro number, respectively, and I_s is the ionic strength in mole/liter. For $T = 298.15$ and $I_s = 0.1$, we can get

$$
(3)\qquad \alpha = 7042.940010604046, \qquad \kappa^2 = 0.8482715968170331,
$$

which will be used in our numerical tests.

To overcome the difficulties brought by Dirac distribution sources $\delta_{\mathbf{r}_j}$, exponential nonlinear term $\sinh(\cdot)$ and the discontinuous permittivity coefficients, variants of solution decomposition techniques have been proposed, see [4, 21, 28, 34] for instances. It is worth mentioning that no matter which solution decomposition method is adopted, the goal is to guarantee accuracy of the numerical solution by isolating the singularities, although different solution schemes lead to distinct formulations of equations. Here following [28, 34], we split the solution u of (1) into the form

$$
(4)\qquad u = G + \Psi + \tilde{\Phi},
$$

where G is given by the expression

$$G(\mathbf{r}) = \frac{\alpha}{4\pi\epsilon_p}\sum_{j=1}^{n_p}\frac{z_j}{|\mathbf{r}-\mathbf{r}_j|}, \tag{5}$$

Ψ is a solution of the linear interface boundary value problem

$$\begin{cases} \Delta\Psi(\mathbf{r}) = 0, & \mathbf{r}\in D_p\cup D_s,\\ \Psi(\mathbf{s}^+) = \Psi(\mathbf{s}^-), & \mathbf{s}\in\Gamma,\\ \epsilon_s\dfrac{\partial\Psi(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_p\dfrac{\partial\Psi(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} + (\epsilon_p-\epsilon_s)\dfrac{\partial G(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})}, & \mathbf{s}\in\Gamma,\\ \Psi(\mathbf{s}) = g(\mathbf{s}) - G(\mathbf{s}), & \mathbf{s}\in\partial\Omega, \end{cases} \tag{6}$$

and $\tilde{\Phi}$ is a solution of the nonlinear interface boundary value problem

$$\begin{cases} \Delta\tilde{\Phi}(\mathbf{r}) = 0, & \mathbf{r}\in D_p,\\ -\epsilon_s\Delta\tilde{\Phi}(\mathbf{r}) + \kappa^2\sinh(G+\Psi+\tilde{\Phi}) = 0, & \mathbf{r}\in D_s,\\ \tilde{\Phi}(\mathbf{s}^+) = \tilde{\Phi}(\mathbf{s}^-), \quad \epsilon_s\dfrac{\partial\tilde{\Phi}(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_p\dfrac{\partial\tilde{\Phi}(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})}, & \mathbf{s}\in\Gamma,\\ \tilde{\Phi}(\mathbf{s}) = 0, & \mathbf{s}\in\partial\Omega. \end{cases} \tag{7}$$

Here $\dfrac{\partial G(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})} = \nabla G\cdot\mathbf{n}$ with ∇G being given by

$$\nabla G(\mathbf{r}) = -\frac{\alpha}{4\pi\epsilon_p}\sum_{j=1}^{n_p} z_j\frac{\mathbf{r}-\mathbf{r}_j}{|\mathbf{r}-\mathbf{r}_j|^3}. \tag{8}$$

As pointed out in [28], the above solution decomposition is a natural way to split the electrostatic potential in Physics. The three component functions G, Ψ, and $\tilde{\Phi}$ can be regarded as the electrostatics induced by atomic charges, interface and boundary conditions, and ionic charges, respectively. Meanwhile, as shown in [29], because G contains all the singular points of u, both Ψ and $\tilde{\Phi}$ become well-defined continuous functions without any singularity. It is also noted that G is given explicitly, while Ψ and $\tilde{\Phi}$ are implicitly determined by the linear boundary value problem (6) and nonlinear problem (7), respectively. The main task of the rest of paper is to provide the appropriate numerical schemes for (6) and (7) such that the sufficient accurate approximations of $\Theta = \Psi + \tilde{\Phi}$ and PBE solution u can be obtained.

3. The finite volume element methods for solving PBE

Let $\mathcal{T}_h$ be a *conformal* triangulation of Ω. In terms of $\mathcal{T}_h$, we define a finite element space

$$U_h := \left\{v\in C(\overline{\Omega}) : v|_\tau\in\mathbf{P}_1, \text{for all}\tau\in\mathcal{T}_h\right\} \tag{9}$$

where $\mathbf{P}_1$ is the set of all polynomials of degree equal or less than 1. It is not difficult to see that U_h is a subspace of the usual Sobolev function space $H^1(\Omega)$. Now let $\mathcal{D}_h$ be another partition of Ω. The element V of $\mathcal{D}_h$ is called *control volume* and is chosen as a polygon as usual (see [33] for the construction of the control volume for each vertex). Let V_h be a piecewise constant function space in terms of $\mathcal{D}_h$ defined by

$$V_h := \left\{v\in L_2(\overline{\Omega}) : v|_V = \text{constant} \ \text{ for all } V\in\mathcal{D}_h\right\}.$$

We call U_h the *trial space*, and V_h the *test space*. It is easy to see that $\dim U_h = \dim V_h$.

As we have seen, the Ψ equation (6) has the non-homogeneous flux condition on the interface Γ (i.e., $[\epsilon(\mathbf{s})\frac{\partial\Psi(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})}] \neq 0$). To handle the non-homogeneous condition, we propose a control-volume decomposition technique as an extension of finite volume methods. That is, as shown in Figure 2, for any vertex on the interface Γ determined by the line segment AB and AC, we actually separate the control volume (i.e., the polygon determined by the black real segments) into two small ones, which are the intersection of the control volume with the polygon $EBAC$ and $BACGF$, respectively. It is clear that the *triangulation* of the interface Γ is simply the common boundary of these two small control volumes. As a result, we are able to proceed as usual to apply the integration-by-part on each decomposed small control volume so as to incorporate the flux-jump condition in the variational form.

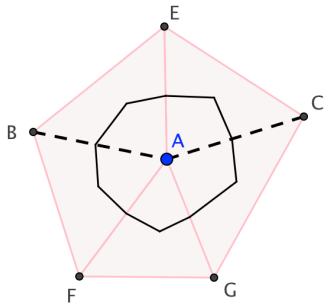

FIGURE 2. The decomposition of the control volume for the vertex A on the interface Γ.

Based on the above new technique, we have the following variational form to obtain a finite volume element approximation of the Ψ equation: Find $\psi_h \in U_h$ with $\psi_h|_{\partial\Omega} = I_h(g - G)$ such that

$$a_h(\psi_h, v) = -(\epsilon_p - \epsilon_s)\int_\Gamma \frac{\partial G}{\partial \mathbf{n}} v d\mathbf{s} \quad \forall v \in V_h^0, \tag{10}$$

where $I_h : H^1 \to U_h$ is the classic Lagrange interpolating operator, the finite volume bilinear form $a_h(v, v_h)$ for all $v \in H^1(\Omega), v_h \in V_h$ is defined by

$$a_h(v, v_h) := -\epsilon_p \sum_{E\in\mathcal{E}_h} \int_{E\cap D_p} \frac{\partial v}{\partial \mathbf{n}}[v_h] d\mathbf{s} - \epsilon_s \sum_{E\in\mathcal{E}_h} \int_{E\cap D_s} \frac{\partial v}{\partial \mathbf{n}}[v_h] d\mathbf{s} \tag{11}$$

with $\mathcal{E}_h$ being the set of all 2-d faces in $\mathcal{D}_h$, $\mathbf{n}$ the unit normal direction on the face E and $[v_h]$ the jump of v_h on E, and V_h^0 is a subspace of V_h defined by

$$V_h^0 = \{v \in V_h \mid v = 0 \text{ on } \partial\Omega\}.$$

Once ψ_h is available, we can compute the finite volume element approximation of $\tilde{\Phi}$ by seeking ϕ_h in

$$U_h^0 = \{w_h \in U_h \mid w_h = 0 \text{ on } \partial\Omega\}$$

such that

$$a_h(\phi_h, v) = -\int_{D_s} \kappa^2 \sinh(G + \psi_h + \phi_h) v d\mathbf{x} \quad \forall v \in V_h^0. \tag{12}$$

Since (12) is a nonlinear system, we solve it by the Newton iteration as follows: for $k = 0, 1, 2, \ldots$, let

$$\phi_h^{(k+1)} = \phi_h^{(k)} + p_k, \tag{13}$$

where $\phi_h^{(0)}$ is an initial guess, $p_k \in U_h^0$ is computed by solving the linear system:

$$b_h(p_k, v) = -a_h(\phi_h^{(k)}, v) - \kappa^2 \int_{D_s} \sinh(G + \psi_h + \phi_h^{(k)}) v d\mathbf{x}, v \in V_h^0 \tag{14}$$

with the bilinear form

$$b_h(p, v) = a_h(p, v) + \int_\Omega \tilde{\kappa}^2 \cosh(G + \psi_h + \phi_h^{(k)}) p v d\mathbf{x} \quad \forall v \in V_h^0,$$

where $\tilde{\kappa} = \kappa$ in D_s and vanishes in D_p. We stop the Newton iterations until the following termination rule is met

$$\left\| -a_h(\phi_h^k, v) - \kappa^2 \int_{D_s} \sinh(G + \psi_h + \phi_h^k) v d\mathbf{x} \right\| < tol, \tag{15}$$

where $\|\cdot\|$ is the Euclidean norm and *tol* is set to 10^{-7} by default.

3.1. The finite volume element PBE solver. Based on the above reformulation, we are in position to solve PBE using the proposed finite volume element method. For clarity, we describe it in Algorithm 1.

Algorithm 1 (The finite volume element PBE solver). *Suppose the domain Ω is sufficiently large. A solution u of PBE is approximated in the following five steps:*

1. *Construct an interface-matched tetrahedral mesh for a given protein.*
2. *Calculate G and its gradient ∇G on Ω according to* (5) *and* (8)*, respectively.*
3. *Solve the linear variational system* (11) *for Ψ.*
4. *Solve the nonlinear problem* (12) *of $\tilde{\Phi}$ by the Newton iterations with the termination rule* (15)*.*
5. *Construct u by the solution decomposition $u = G + \Psi + \tilde{\Phi}$.*

To construct the interface-matched meshes, we adopt the revised version of the molecular surface and volumetric mesh generation program package GAMer [32, 34] to be the mesh generator, where the smooth Gaussian surface is set as the default to define the molecular surface, while two other choices are available: the solvent-accessible surface and the solvent-excluded surface. We write our own codes to assemble the stiff matrix and the right-hand side vector so that we can easily handle the non-homogeneous flux condition occurred on Γ for Ψ equation. For the assembling process, same as the finite element method, all cells of the mesh are iterated one by one and on each cell, the local stiffness matrix and vector are assembled by mapping to the reference one and then added to the global stiffness matrix and vector correspondingly.

By a simplification technique shown in [33], for only $a_h(v, v_h)$, the assembled stiffness matrix of the FVEM is identical to that of the FEM, which implies that we could fully use the same mesh, data structure and even some mutual existed library of the FEM to assemble the FVEM stiffness matrix. For this reason, we fulfilled Algorithm 1 based on the state-of-the-art finite element library `DOLFIN` [18] from the `FEniCS` project. In order to speedup the assemble process, we programed all the functions needed for assembling in `Fortran`. Meanwhile, to accelerate the computation, all heavy computations were carried out also in `Fortran` subroutines. Using the Fortran-to-Python interface generator `f2py`

(*http://cens.ioc.ee/projects/f2py2e/*), we converted all the `Fortran` subroutines to `Python` modules. With these modules, we could write the main function in `Python`.

4. Analysis

To analyze the stability and convergence property of our finite volume scheme for $\Theta = \Psi + \tilde{\Phi}$, we first write the two finite volume schemes (10) and (12) into the following unified one. In fact, by letting $\theta_h = \psi_h + \phi_h$ and summarizing (10) and (12), we have

$$a_h(\theta_h, v) + (\tilde{\kappa}^2 \sinh(G + \theta_h), v) = -(\epsilon_p - \epsilon_s) \int_\Gamma \frac{\partial G}{\partial \mathbf{n}} v d\mathbf{s} \quad \forall v \in V_h^0. \tag{16}$$

Since the trial space U_h is different from the test space V_h, the finite volume scheme (16) is a Petrov-Galerkin scheme which is difficult to be analyzed. Fortunately, this Petrov-Glerkin scheme can be transformed into a Galerkin scheme by introducing the following from-trial-to-test space mapping. Let N_h be the set of all vertices of $\mathcal{T}_h$. For each $P \in \mathcal{N}_h$, let ϕ_P the usual nodal basis function satisfying

$$\phi_P \in U_h, \text{and } \phi_P(P') = \delta_{P,P'}, \forall P' \in \mathcal{N}_h,$$

and let ψ_P be the characteristic function on the control volume of vertex P. Then $U_h = \text{span}\{\phi_P : P \in N_h\}$ and $V_h = \text{span}\{\psi_P : P \in N_h\}$. The linear bijective mapping $P_h : U_h \to V_h$ is defined by letting

$$P_h(\sum_{P \in \mathcal{N}_h} c_P \phi_P) = \sum_{P \in \mathcal{N}_h} c_P \psi_P, \forall c_P \in \mathbf{R}, P \in \mathcal{N}_h,$$

and we denote $v_h^* = P_h v_h, \forall v_h \in U_h$. With this mapping, (10) can be rewritten as the Galerkin scheme

$$a_h(\theta_h, v^*) + (\tilde{\kappa}^2 \sinh(G + \theta_h), v^*) = -(\epsilon_p - \epsilon_s) \int_\Gamma \frac{\partial G}{\partial \mathbf{n}} v^* d\mathbf{s} \quad \forall v \in U_h^0. \tag{17}$$

Next we analyze the Algorithm 1 by using the Galerkin scheme (17). It is known that (see e.g. [33]) that the identity

$$a_h(v_h, w_h^*) = \epsilon_p \int_{D_p} \nabla v_h \cdot \nabla w_h d\mathbf{x} + \epsilon_s \int_{D_s} \nabla v_h \cdot \nabla w_h d\mathbf{x}, \forall v_h, w_h \in U_h \tag{18}$$

holds. As direct consequences, we have the coercivity

$$a_h(v_h, v_h^*) \geq \min\{\epsilon_p, \epsilon_s\} |v_h|_1^2, \forall v_h \in U_h,$$

where $|\cdot|_1$ is the standard Sobolev H^1 seminorm. And we also have the continuity

$$a_h(v_h, w_h^*) \leq \max\{\epsilon_p, \epsilon_s\} \|v_h\|_1 \|w_h\|_1, \forall v_h, w_h \in U_h.$$

On the other hand, the function sinh is monotone increasing in the sense that

$$\big(\tilde{\kappa}^2(\sinh(G + v_1) - \sinh(G + v_2)), v_1 - v_2\big) \geq 0, \forall v_1, v_2 \in H^1,$$

and bounded in the sense that for $v_1, v_2 \in L^\infty(\Omega)$,

$$\|\sinh v_1 - \sinh v_2\|_0 \leq C \|v_1 - v_2\|_0,$$

where $\|\cdot\|_0$ is the standard L^2-norm.

In the following, we explain that the FVEM solution θ_h is uniformly bounded (independent of the mesh size h) in the space $L^\infty(\Omega)$. By observing that the function G is smooth away from $\mathbf{r}_i$, we have the fact that $G \in C^\infty(D_s) \cap C^\infty(\Gamma) \cap C^\infty(\partial\Omega)$. Consequently, both Ψ and $\tilde{\Phi}$ (and consequently $\Theta = \Psi + \tilde{\Phi}$) are in $H^2(D_p) \cap H^2(D_s) \cap H_0^1(\Omega) \cap L^\infty(\Omega)$ (see e.g. [4]). Furthermore, when the interface Γ is sufficiently smooth, we also have $\Theta \in W^{2,\infty}(D_p) \cap W^{2,\infty}(D_s)$. Then by applying

the maximum-norm error estimates techniques developed in [5, 7, 8] to the elliptic interface problem (17), we actually have

$$\|\Theta - \theta_h\|_\infty \lesssim Ch^2(\|\Theta\|_{2,\infty,D_p} + \|\Theta\|_{2,\infty,D_s}),$$

which implies the uniform boundedness of $\|\theta_h\|_\infty$.

By the above uniform boundedness and the coercivity and continuity,

$$\begin{aligned}
\|\theta_h - I_h\Theta\|_1^2 \leq & Ca_h(\theta_h - I_h\Theta, (\theta_h - I_h\Theta)^*) \\
\leq & C\left(a_h(\theta_h - I_h\Theta, (\theta_h - I_h\Theta)^*)\right. \\
& \left. + \left(\tilde{\kappa}^2(\sinh(G+\theta_h) - \sinh(G + I_h\Theta)), \theta_h - I_h\Theta\right)\right) \\
= & C\left(a_h(\Theta - I_h\Theta, (\theta_h - I_h\Theta)^*)\right. \\
& \left. + \left(\tilde{\kappa}^2(\sinh(G+\Theta) - \sinh(G + I_h\Theta)), \theta_h - I_h\Theta\right)\right) \\
\leq & C\|\Theta - I_h\Theta\|_1\|\theta_h - I_h\Theta\|_1. \qquad (19)
\end{aligned}$$

Note that in the above reasoning, C is a constant independent of the mesh size h, but its value may vary from line to line. The inequality (19) implies that

$$\|\theta_h - I_h\Theta\|_1 \leq C\|\Theta - I_h\Theta\|_1$$

and thus by the triangle inequality, we have

$$\|\Theta - \theta_h\|_1 \leq C\|\Theta - I_h\Theta\|_1. \qquad (20)$$

In other words, we have optimal convergence order (1st order here) for the energy norm error $\|\Theta - \theta_h\|_1$.

With the uniform boundedness of $\|\theta_h\|_\infty$, we can also obtain the following L^2-norm error estimate by applying the techniques developed in [6, 9],

$$\|\Theta - \theta_h\|_0 \leq Ch^2. \qquad (21)$$

5. Numerical experiments

In this section, we report the numerical experiments using our new `Python` program of Algorithm 1. For simplicity, we set $\epsilon_p = 2.0$, $\epsilon_s = 80.0$, $T = 298.15$, $I_s = 0.1$, and all the numerical tests were done on one processor of a Macbook Pro with the 2.7 GHz Intel Core I5 and 8 GB memory. Without explicit statement, the boundary condition is set to be zero.

5.1. Local Poisson test model. In this subsection, we will use the following local Poisson test model with available analytical solution reported in [13, 31, 30] to do the validation tests

$$\begin{cases}
-\epsilon_p\Delta u(\mathbf{r}) = \alpha\sum_{j=1}^{n_p} z_j\delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\
-\epsilon_s\Delta u(\mathbf{r}) = 0, & \mathbf{r} \in D_s, \\
u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s\dfrac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p\dfrac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\
u(\mathbf{r}) \to 0, & \mathbf{r} \to \infty,
\end{cases} \qquad (22)$$

where D_p is a spherical domain centered at origin with the radius a, $\mathbf{r}_j$ and z_j are the position and charge number of the jth atom of the protein, respectively, $\partial\Omega$ denotes the boundary of Ω, $\delta_{\mathbf{r}_j}$ is the Dirac delta distribution at point $\mathbf{r}_j$, and $\mathbf{n}(\mathbf{s})$ is the unit outward normal vector of D_p.

By superposition, we have the analytical formula $U(\mathbf{r}) = \alpha\sum_{j=1}^{n_p} z_j u_j(\mathbf{r})$, where $u_j(\mathbf{r})$ denotes the solution of problem (22) with the right-hand function being $\delta_{\mathbf{r}_j}$ in D_p and is given by

(1) If $|\mathbf{r}_j| = 0$:

$$\begin{cases} u_j(\mathbf{r}) = \dfrac{1}{4\pi\epsilon_p|\mathbf{r}|} + \dfrac{1}{4\pi a}(\dfrac{1}{\epsilon_s} - \dfrac{1}{\epsilon_p}), & \mathbf{r} \in D_p, \\ u_j(\mathbf{r}) = \dfrac{1}{4\pi\epsilon_s|\mathbf{r}|}, & \mathbf{r} \in D_s. \end{cases}$$

(2) If $|\mathbf{r}_j| \neq 0$:

$$\begin{cases} u_j(\mathbf{r}) = \dfrac{1}{4\pi\epsilon_p}\dfrac{1}{|\mathbf{r}-\mathbf{r}_j|} + \displaystyle\sum_{n=0}^{\infty} A_{j,n}|\mathbf{r}|^n P_n(\frac{\mathbf{r}_j\cdot\mathbf{r}}{|\mathbf{r}_j||\mathbf{r}|}), & \mathbf{r} \in D_p, \\ u_j(\mathbf{r}) = \displaystyle\sum_{n=0}^{\infty} B_{j,n}|\mathbf{r}|^{-n-1} P_n(\frac{\mathbf{r}_j\cdot\mathbf{r}}{|\mathbf{r}_j||\mathbf{r}|}), & \mathbf{r} \in D_s. \end{cases}$$

Here P_n denotes the Legendre polynomial satisfying the following recursive relationship

$$P_0(x) = 1, \quad P_1(x) = x, \quad (n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x) \quad n \geq 1,$$

and the coefficients $A_{j,n}$ and $B_{j,n}$ are defined by

$$A_{j,n} = \frac{(\epsilon_p - \epsilon_s)(n+1)|\mathbf{r}_j|^n}{4\pi\epsilon_p a^{2n+1}[n\epsilon_p + (n+1)\epsilon_s]}, \qquad B_{j,n} = \frac{(2n+1)|\mathbf{r}_j|^n}{4\pi[n\epsilon_p + (n+1)\epsilon_s]}.$$

For fully verifying our new algorithm, we consider the following artificial constructed problem on $\Omega = [-2,2]^3$:

$$\begin{cases} -\epsilon_p\Delta u(\mathbf{r}) = \alpha\displaystyle\sum_{j=1}^{n_p} z_j\delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_s\Delta u(\mathbf{r}) + \kappa^2\sinh(u(\mathbf{r})) = \kappa^2\sinh(U(\mathbf{r})), & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s\dfrac{\partial u(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_p\dfrac{\partial u(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ g(\mathbf{r}) = U(\mathbf{r}), & \mathbf{r} \in \partial\Omega, \end{cases} \tag{23}$$

which still has the analytical solution $U(\mathbf{r})$. To carry out the numerical tests, we constructed four nested unstructured meshes (i.e., a nested mesh is obtained by uniformly refining a given mesh) on the domain Ω with the interface $\Gamma = \{\mathbf{r} : \|\mathbf{r}\| = 1.0\}$. The number of vertices of these four nested meshes are 3779, 30237, 240377, and 1913841, respectively. Using these four nested unstructured meshes, we calculated the analytical solution of (22) and its numerical solutions to verify our new program. Here the charge numbers z_j and atomic positions $\mathbf{r}_j$ were obtained from a PQR file of a protein with PDB ID 4PTI, which has 892 atoms (i.e., $n_p = 892$), and each atom coordinate $\mathbf{r}_j$ ($j = 1$ to 892) was divided by 28 to rescale it into the unit spherical region D_p so that the structure of the protein is preserved within the unit ball. And we truncated the series to the first 20 terms in order to get the analytical solution.

As before, in order to guarantee the numerical solution's accuracy, we applied the solution decomposition so that $u = G + \Psi + \tilde{\Phi}$, where G is given in (5) and Ψ is determined by (6) with the boundary function being $U - G$. For the nonlinear problem $\tilde{\Phi}$, due to the additional term, we modified the right hand side term of the Newton equation by adding the given term $\kappa^2\int_{D_s}\sinh(U)v$ for $v \in V_h^0$ and used the constant one as the initial guess for the modified Newton method (since $\tilde{\Phi} \equiv 0$). It is easy to see that these small changes do not produce any side effect on our validation. Meanwhile, we know the solution of (23) is proportional to α. To avoid the overflow problem caused by the hyperbolic functions, we set $\alpha = 1.0$ in

this validation test. Furthermore, since G is analytically computed, we only report the errors for Ψ and $\tilde{\Phi}$. All the numerical results are reported in Table 1.

TABLE 1. Numerical results for the local Poisson test model (22).

Mesh index	Mesh size h	Relative Error $\frac{\|\Psi_h+\tilde{\Phi}_h-\Psi-\tilde{\Phi}\|_0}{\|\Psi+\tilde{\Phi}\|_0}$	Convergence order	Newton Iteration
1	0.921	2.02×10^{-2}	–	3
2	0.496	5.09×10^{-3}	2.23	3
3	0.280	1.28×10^{-3}	2.41	3
4	0.146	3.24×10^{-4}	2.11	3
Mesh index	Mesh size h	Relative Error $\frac{\|\Psi_h+\tilde{\Phi}_h-\Psi-\tilde{\Phi}\|_1}{\|\Psi+\tilde{\Phi}\|_1}$	Convergence order	
1	0.921	4.47×10^{-2}	–	
2	0.496	1.62×10^{-2}	1.64	
3	0.280	6.17×10^{-3}	1.69	
4	0.146	2.84×10^{-3}	1.19	
Mesh index	Mesh size h	Relative Error $\frac{\|\Psi_h+\tilde{\Phi}_h-\Psi-\tilde{\Phi}\|_\infty}{\|\Psi+\tilde{\Phi}\|_\infty}$	Convergence order	
1	0.921	4.44×10^{-2}	–	
2	0.496	1.99×10^{-2}	1.30	
3	0.280	5.15×10^{-3}	2.36	
4	0.146	1.83×10^{-3}	1.59	

From Table 1, we can see for solving the the non-homogeneous interface problem (e.g., (6) of Ψ), the proposed finite volume element method has the convergence order around 2 under the l_2 norm and around 1 under the H^1 norm, well matching the theoretical results and thus validating our program. Here on the 4th mesh, the number of computed terms in the series was increased to 30 so that the discretization error would not be the dominated one. Under the l_∞ norm, the convergence order is reduced as expected. Meanwhile, it can be also observed that on these four nested meshes, Newton method shows the local quadratic convergence property as well, and surprisingly exact the same Newton iterations are taken.

5.2. Predicting electrostatic solvation free energies of proteins. As pointed out in various literatures, one important application of implicit continuum models is to predict the electrostatic solvation free energy of a biomolecule. According to the solution decomposition (4), the electrostatic solvation energy E can be estimated by the formula

$$E = \frac{N_A}{4184} \cdot \frac{k_B T}{2} \sum_{j=1}^{n_p} z_j \left(\Psi(\mathbf{r}_j) + \tilde{\Phi}(\mathbf{r}_j) \right) \quad \text{in kilocalorie per mole (kcal/mol).} \tag{24}$$

In calculation, the numerical solutions Ψ_h and $\tilde{\Phi}_h$ are obtained from solving (6) and (7). As shown in [34], this quantity has been used to show the numerical stability of the finite element solver. Here we could fully use this quantity to further verify our new program by comparing the computed electrostatic solvation energies using the finite volume element method and finite element method, which are labeled as E_{FVEM} and F_{FE}, respectively.

To do this, we use the coarsest finite element mesh sets used in [34] for four proteins with PDB ID 2LZX, 1SVR, 1CID and 2AQ5, respectively. The PDB files of these proteins were downloaded from the PDB website and then converted to PQR files by the tool `PDB2PQR` [25] with the `CHARMM` force field. These four proteins have 488, 1433, 2783, and 6024 atoms, respectively. Here, each molecular surface (i.e., the interface Γ) was generated by our modified version of `GAMer` based on the Gaussian blurring approach [32]. It has been shown in [34] that the generated Gaussian surface can be close to a molecular structure, and meanwhile its molecular shape is close to the ones generated using definitions of both the solvent-accessible surface and solvent-excluded surface. Therefore, here we use the Gaussian surface definition for simplicity since it generates the smoother molecular surface. The numerical results are reported in Table 2.

TABLE 2. Electrostatic solvation free energies obtained using different numerical methods.

PDB ID	Number of mesh nodes	E_{FE}	E_{FVEM}
2LZX	11034	-214.05	-213.47
1SVR	25842	-288.86	-286.03
1CID	32940	-313.78	-310.23
2AQ5	52566	-903.92	-897.48

From Table 2, we see that with more atoms, more unstructured mesh nodes it has. And the electrostatic solvation free energies predicted by the finite element solver and the new finite volume element solver are highly close, which also matches the common sense and further validates our new program. Since the finite element solver has been shown to be stable, it is for sure that the new finite volume element PBE solver is also numerical stable. And all the applications that have been done using the finite element PBE solvers in [28, 34] can be repeated here using the new program package. For simplicity, it is omitted in this paper.

Acknowledgments

The authors thank the anonymous referees for the suggestions to improve the paper's overall quality. The research of Dr. Ying was supported by the National Natural Science Foundation of China (Grant No. 11701576) and the Natural Science Foundation of Hunan Province (Grant No. 2019JJ50786). The research of Dr. Zou was supported in part by the special project High Performance Computing of National Key Research and Development Program 2016YFB0200604, NSFC Grant 12071496, Guangdong Provincial NSF Grant 2017B030311001, and by Guangdong Province Key Laboratory of Computational Science at the Sun Yat-sen University (2020B1212060032).

References

[1] N. Baker, M. Holst, F. Wang, Adaptive multilevel finite element solution of the Poisson-Boltzmann equation ii. refinement at solvent-accessible surfaces in biomolecular systems, Journal of Computational Chemistry 21 (15) (2000) 1343–1352.

[2] A. H. Boschitsch, M. O. Fenley, Hybrid boundary element and finite difference method for solving the nonlinear Poisson-Boltzmann equation, Journal of Computational Chemistry 25 (7) (2004) 935–955.

[3] Z. Cai, On the finite volume element method, Numerische Mathematik 58 (1990) 713–735.

[4] L. Chen, M. J. Holst, J. Xu, The finite element approximation of the nonlinear Poisson-Boltzman equation, SIAM Journal on Numerical Analysis 45 (6) (2007) 2298–2320.

[5] S. H. Chou, Q. Li, Error estimates in L_2, H^1 and L^∞ in covolume methods for elliptic and parabolic problems: a unified approach, Mathematics of Computation 69 (2000) 103–120.
[6] R. E. Ewing, T. Lin, Y. Lin, On the accuracy of the finite volume element method based on piecewise linear polynomials, SIAM Journal on Numerical Analysis 39 (2002) 1865–1888.
[7] W. He, Z. Zhang, Q. Zou, Maximum-norms error estimates for high order finite volume schemes over quadrilateral meshes, Numerische Mathematik 136 (2018) 473–500.
[8] W. He, Z. Zhang, Q. Zou, Local superconvergence of post-processed high-order finite volume element solutions, Advances in Computational Mathematics 46 (2020) 1–26.
[9] Y. Lin, M. Yang, Q. Zou, L_2 error estimates for a class of any order finite volume schemes over quadrilateral meshes, SIAM Journal on Numerical Analysis 53 (4) (2015) 2030–2050.
[10] M. J. Holst, F. Saied, Numerical solution of the nonlinear Poisson–Boltzmann equation: developing more robust and efficient methods, Journal of Computational Chemistry 16 (3) (1995) 337–364.
[11] R. Li, Z. Chen, W. Wu, The generalized difference methods for partial differential equations, Marcel Dikker, New York, 2000.
[12] J. Li, D. Xie, An effective minimization protocol for solving a size-modified Poisson-Boltzmann equation for biomolecule in ionic solvent, International Journal of Numerical Analysis and Modeling 12 (2) (2015) 286–301.
[13] J. Li, J. Ying, B. Lu, A flux-jump preserved gradient recovery technique for accurately predicting the electrostatic field of an immersed biomolecule, Journal of Computational Physics 396 (2019) 193–208.
[14] J. Li, J. Ying, D. Xie, On the analysis and application of an ion size-modified Poisson-Boltzmann equation, Nonlinear Analysis: Real World Applications 47C (2019) 188–203.
[15] J. Li, J. Ying, A simple and efficient technique to accelerate the computation of a nonlocal dielectric model for electrostatics of biomolecule, Journal of Industrial & Management Optimization 16 (1) (2020) 357–369.
[16] F. Liebau, The finite volume element method with quadratic basis function, Computing 57 (1996) 281–299.
[17] E. R. Lima, F. W. Tavares, E. C. Biscaia Jr, Finite volume solution of the modified Poisson-Boltzmann equation for two colloidal particles, Physical Chemistry Chemical Physics 9 (24) (2007) 3174–3180.
[18] A. Logg, G. N. Wells, DOLFIN: automated finite element computing, ACM Transactions on Mathematical Software (TOMS) 37 (2) (2010) 1–28.
[19] B. Lu, X. Cheng, J. Huang, J. A. McCammon, AFMPB: an adaptive fast multipole Poisson–Boltzmann solver for calculating electrostatics in biomolecular systems, Computer Physics Communications 181 (6) (2010) 1150–1160.
[20] B. Lu, M. J. Holst, J. A. McCammon, Y. Zhou, Poisson–Nernst–Planck equations for simulating biomolecular diffusion–reaction processes I: finite element solutions, Journal of Computational Physics 229 (19) (2010) 6979–6994.
[21] B. Lu, Y. Zhou, M. Holst, J. McCammon, Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications, Communications in Computational Physics 3 (5) (2008) 973–1009.
[22] M. T. Neves-Petersen, S. B. Petersen, Protein electrostatics: a review of the equations and methods used to model electrostatic equations in biomolecules–applications in biotechnology, Biotechnology Annual Review 9 (2003) 315–395.
[23] Z. Schuss, B. Nadler, R. S. Eisenberg, Derivation of Poisson and Nernst-Planck equations in a bath and channel from a molecular model, Physical Review E 64 (3) (2001) 036116.
[24] S. Tomac, A. Gräslund, Multi-multigrid solution of modified Poisson-Boltzmann equation for arbitrarily shaped molecules, Journal of Computational Chemistry 19 (8) (1998) 893–901.
[25] S. Unni, Y. Huang, R. M. Hanson, M. Tobias, S. Krishnan, W. W. Li, J. E. Nielsen, N. A. Baker, Web servers and services for electrostatics calculations with APBS and PDB2PQR, Journal of Computational Chemistry 32 (7) (2011) 1488–1491.
[26] C. L. Vizcarra, S. L. Mayo, Electrostatics in computational protein design, Current Opinion in Chemical Biology 9 (6) (2005) 622–626.
[27] J. Wang, Q. Cai, Z.-L. Li, H.-K. Zhao, R. Luo, Achieving energy conservation in Poisson–Boltzmann molecular dynamics: accuracy and precision with finite-difference algorithms, Chemical Physics Letters 468 (4) (2009) 112–118.
[28] D. Xie, New solution decomposition and minimization schemes for Poisson-Boltzmann equation in calculation of biomolecular electrostatics, Journal of Computational Physics 275 (2014) 294–309.

[29] D. Xie, J. Li, A new analysis of electrostatic free energy minimization and Poisson-Boltzmann equation for protein in ionic solvent, Nonlinear Analysis: Real World Applications 21 (2015) 185–196.
[30] D. Xie, H. W. Volkmer, J. Ying, Analytical solutions of nonlocal poisson dielectric models with multiple point charges inside a dielectric sphere, Physical Review E 93 (4) (2016) 043304.
[31] D. Xie, J. Ying, A new box iterative method for a class of nonlinear interface problems with application in solving Poisson–Boltzmann equation, Journal of Computational and Applied Mathematics 307 (2016) 319–334.
[32] Y. Xie, J. Ying, D. Xie, SMPBS: web server for computing biomolecular electrostatics using finite element solvers of size modified Poisson-Boltzmann equation, Journal of Computational Chemistry 38 (8) (2017) 541–552.
[33] J. Xu, Q. Zou, Analysis of linear and quadratic simplicial finite volume methods for elliptic equations, Numerische Mathematik 111 (3) (2009) 469–492.
[34] J. Ying, D. Xie, A new finite element and finite difference hybrid method for computing electrostatics of ionic solvated biomolecule, Journal of Computational Physics 298 (2015) 636–651.
[35] J. Ying, D. Xie, A hybrid solver of size modified Poisson-Boltzmann equation by domain decomposition, finite element, and finite difference, Applied Mathematical Modelling 58 (2018) 166–180.
[36] J. Ying, D. Xie, An accelerated nonlocal Poisson-Boltzmann equation solver for electrostatics of biomolecule, International Journal for Numerical Methods in Biomedical Engineering 34 (11) (2018) e3129.
[37] Z. Zhang, Q. Zou, Vertex-centered finite volume schemes of any order over quadrilateral meshes for elliptic boundary value problems, Numerische Mathematik 130 (2015) 363–393.
[38] Y. Zhou, S. Zhao, M. Feig, G.-W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, Journal of Computational Physics 213 (1) (2006) 1–30.

School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China.

School of Mathematics and Statistics, HNP-LAMA, Central South University, Changsha, Hunan 410083, China.
E-mail: ying48882345@gmail.com

School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China.